Splunk Assessment of Mitigation Implementations

User Instruction Manual

Author: CDR&T

12/30/2015

This System Design Document outlines the Splunk Assessment of Mitigation Implementations (SAMI) Splunk application, which visualizes Metrics and Measures on network endpoints, providing ease in analysis of mitigation implementations.

Table of Contents

Introduction

Purpose

The purpose of this document is to provide an overview of the Splunk Assessment of Mitigation Implementations (SAMI) application. The following pages provide a high level overview of the tool, design decisions, and benefits, and outline the limitations the tool possesses.

Background

Mitigation implementations have not traditionally been evaluated or prioritized. NSA's Information Assurance Directorate recommends mitigation strategies but there are few methods to validate proper implementation and prioritize fixes. The SAMI application was developed to monitor the degree to which specific aspects of the IAD Top 10 Mitigation Strategies[1] have been deployed on Windows endpoints. It monitors data related to the implementation of specific mitigations and returns prioritized recommendations to more completely implement those recommendations. The application can be used to determine a network's mitigation implementation status and can be monitored over time to demonstrate improvements and identify changes that negatively impact mitigations.

SAMI evaluates several metrics:

- Modern Operating System (MOS)
- Anti-Virus File Reputation Service (AVFRS)
- Host Intrusion Prevention System (HIPS)
- Application Whitelisting (AW)
- Anti-Exploitation (AE)
- Pass-the-Hash (PtH)

Vision

The application aims to make automated measurement of mitigations both common and understandable. More specifically, the goals for SAMI are to:

- Establish the importance of measuring mitigations;
- Jumpstart community discussion on the topic;
- Motivate vendors to support and build upon the capabilities;
- Establish the concept and process of scoring mitigations as part of vulnerability risk scoring

[1] https://www.iad.gov/iad/library/ia-guidance/iads-top-10-information-assurance-mitigation-strategies.cfm

High-Level Description

System Diagram

The following diagram illustrates the main components of SAMI. The Splunk Universal Forwarder is used to run scripts and monitor specific data on Windows endpoints. The data is sent to the indexer to support regularly scheduled evaluations of the data. The evaluated data is displayed in a summary dashboard.

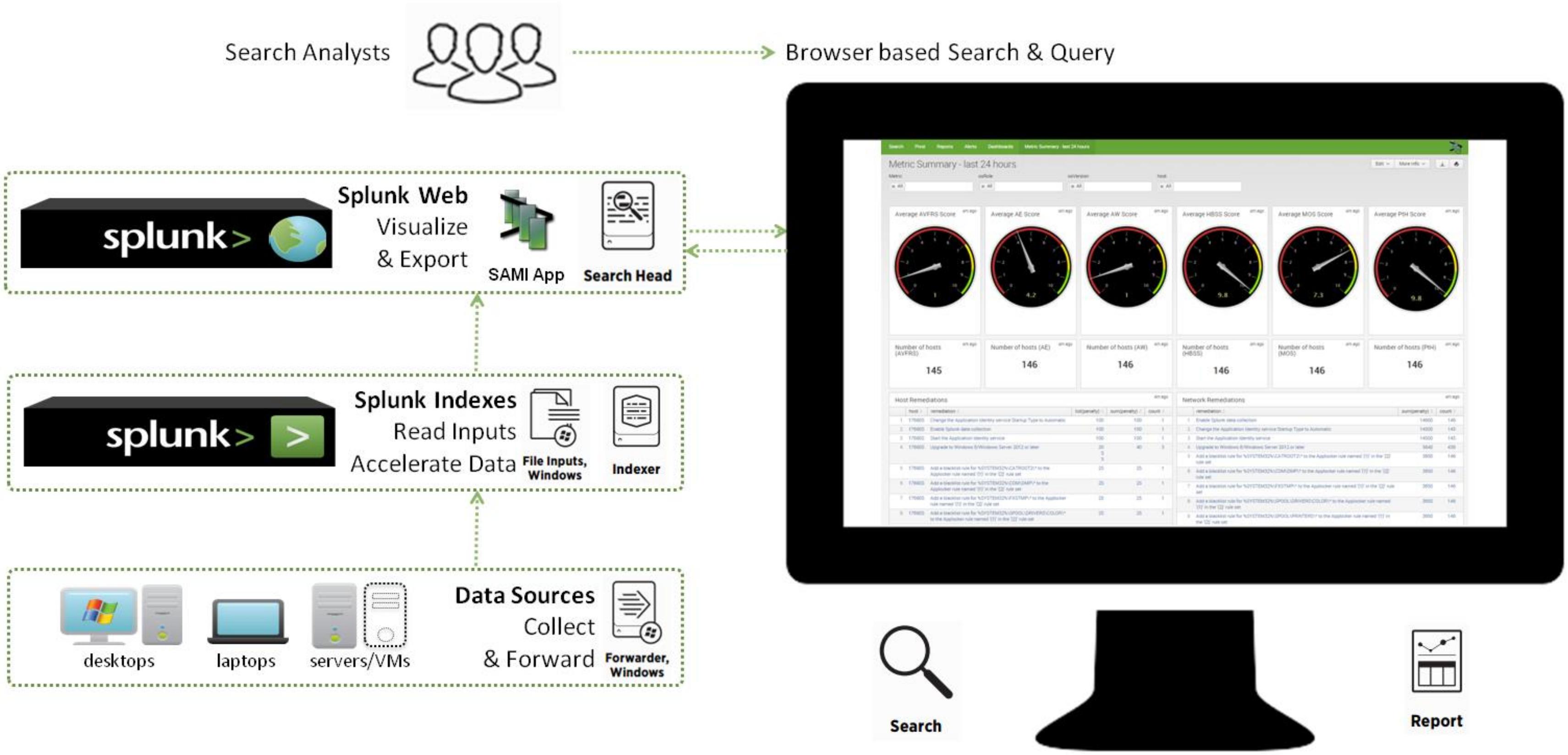

Figure 1 Abstract System Diagram

Application Overview

The SAMI application is divided into two pieces: a Splunk app (SAMI) and a Splunk deployment app (TA-SAMI). The deployment app includes the necessary configuration files and scripts for collecting data on each endpoint. The Splunk app contains the lookups, searches, and dashboards for the end user.

Scoring

Hosts are scored from 1 (worst) to 10 (best) for each metric, based on the penalties applied for each measure in a given metric. Penalties are fixed for each measure but vary from 2 (least penalty) to 100 (worst penalty) based on the severity of the measure. For example, having DEP completely disabled on a host would result in a higher penalty than if one application had opted out of DEP on that host. High penalties result in low scores.

Data

The SAMI application requires data from a variety of sources to perform its measurements. The data sources required for each metric are listed below:

- Modern Operating System (MOS)
    - WMI
- Anti-Virus File Reputation Service (AVFRS)
    - Splunk's WinHostMon
    - Windows Registry
    - Script to query DNS from endpoints
- Host Intrusion Prevention System (HIPS)
    - Windows Registry
    - WMI
- Application Whitelisting (AW)
    - Windows Registry
    - Splunk's WinHostMon
- Anti-Exploitation (AE)
    - Windows Registry
    - WMI
    - Scripts to collect hardware, BIOS, DEP, and null page mapping information
- Pass-the-Hash (PtH)
    - Windows Registry
    - List of Domain Administrators (obtained using Powershell)
    - Splunk Security Event Logs

Application Components

The application consists of a number of components: searches to determine the penalties for each measure, a summary index populated by the searches, a dashboard to visualize the information, and lookups to correlate static information.

Searches

In general, there is one saved search for each measure with some searches including penalties for multiple measures where it made sense (e.g., AVFRS/DAT_OUTDATED and AVFRS/DAT_VERY_OUTDATED are together). The searches analyze information from the appropriate data sources and, if necessary, penalize each host based on the values in the penalty lookup file. The searches are scheduled to run once a day but the specific schedule should be configured for the user's convenience.

In addition to searches for specific measures within a metric, there is also a search for each metric that penalizes hosts if they are missing any measures within that metric. This is done by checking the list of active hosts against the measures the summary index has for each host. For example, if the summary index has information for all AVFRS measures for a host except DAT_OUTDATED, that host is given a penalty of 100 for AVFRS. There is one search per metric, with a title "{metric}_MISSING_DATA", that is scheduled to run daily after all other searches for that metric.

Some of the lookup files are also populated by searches. See the Lookups section for details.

Indexes

The sami_summary index is a summary index used to store the results of each measure's penalty search. A summary index was chosen to speed up the searches on the SAMI dashboard as many of the penalty searches look across all time and can take minutes to run. Additionally, this method prevents the need for a large number of joins when scoring across an entire metric.

Add index 'sami_summary' to the default searchable indexes by going to Splunk Settings -> Access controls -> Role -> admin -> Indexes searched by default and adding 'sami_summary' into the list of default search indexes.

The TA-SAMI app produces registry data, WinHostMon data, WMI data, and script results. This data is stored in the default index with the configuration packaged with the app, but some users may want to put this data in dedicated indexes.

Dashboard

The dashboard provides users with a look at the mitigation status of their network (in total and by host) and recommended remediations sorted by most prevalent failed measure. The dashboard includes charts and tables that show various views of the data:

- Scores by host (with osVersion and role information)
- Prioritized list of remediations per host and per network
- Average score per metric and the number of hosts used in the calculation
- Average total score (note that the total average score is not the average of the component metrics)
- Average scores trended over time
- Average sum of penalties over time
- Host status (active, missing – with or without past data in Splunk)

- Counts of hosts with failed results for each measure
- Actual findings per host for each measure

Lookups

The application uses a number of lookup files for values that do not change often. Below is a description of each lookup file and how it is populated.

- AW_Category_Skip_lookup.csv
    - Populated by AW_Skip_Category_lookup search daily
    - Used to determine whether a given host should skip either AppLocker or SRP measures in the AW metric. It is expected that only one AW solution will be implemented. The method that scores worse is discarded. This lookup records the decision for use in scoring.
    - Fields: host, skip_aw_cat
- AppLocker-blacklistPath.csv
    - Manually generated; not expected to change
    - Used to determine blacklisted paths for AW/AppLocker measures.
    - Fields: BlacklistedPath, Value
- OS-CurrentVersion.csv
    - Manually generated; updated in future versions as new OS's require different check logic
    - Maps operating system version numbers from Windows Registry/WMI to human-readable equivalent (i.e. '6.3'->'Windows 8.1/Windows Server 2012 R2')
    - Fields: CurrentVersion, OS
- SRP-blacklistPath.csv
    - Manually generated; not expected to change
    - Used to determine blacklist paths for AW/SRP measures.
    - Fields: Value, BlacklistedPath
- SRP-defaultExeTypes.csv
    - Manually generated; not expected to change
    - Used to determine types of executables for AW/SRP measures.
    - Fields: Type, Value
- SRP-whitelistPath.csv
    - Manually generated; not expected to change
    - Used to determine whitelist paths for AW/SRP measures.
    - Fields: Value, WhitelistedPath
- hostList.csv
    - Populated by hostList_lookup search daily
    - Used to determine all of the hosts on the network; this is based on hosts active in AD
    - Fields: host
- os_lookup.csv
    - Populated by OS_LOOKUP search weekly

    - Used to determine, for each host, the operating system version, service pack, and architecture (32- or 64-bit), as well as the role of the machine (workstation or server) and the last time it was updated.
    - Fields: host, osVersion, ServicePack, osRole, osArch, lastUpdate
- ruleSet_lookup.csv
    - Manually generated; not expected to change
    - Used to determine the rule sets to check in AW AppLocker measures.
    - Fields: ruleSet
- sami_fix_lookup.csv
    - Manually generated
    - Used to determine the remediation for a given measure. Updating the remediations in this file will automatically update the remediations shown on the SAMI dashboard for the corresponding measure.
    - Fields: id2, measure, metric, remediation
- sami_penalties_lookup.csv
    - Manually generated
    - Used to determine the penalty for a given measure. Updating the penalty values in this file will update the penalties applied by the corresponding measure search the next time it runs. Changing these values is also one way to change the priority of fixes.
    - Fields: metric, category, measure, description, penalty, reason

Additional details about the application can be found in Appendix B.

Limitations

SAMI was written and tested using Splunk 6.2, requires Splunk Universal Forwarders on all endpoints, and only applies to Windows operating systems (up to and including Windows 10).

Maintenance

The SAMI app will require some regular maintenance. Measures that depend on versions of the OS, an application, a patch, or a file need to be updated when new versions are released. Similarly, when new versions of related tools like EMET, HIPS, AppLocker, SRP, and even operating systems are released the measures may need to be re-evaluated.

Some of these updates will require changes to the measure searches and possibly the logic behind them. However, for those likely to change most often, SAMI uses macros to manage the version numbers. These values can be updated through the “Advanced Search > Search Macros” menu by clicking on the macro name and changing the version numbers to the most recent required and/or recommended numbers:

1. get_hips_version
    a. Contains both required and recommended version numbers
2. get_required_av_engine_version

   a. Contains only required version number, but split between two fields (major and minor)
   b. Ex. Required version is 5700.000; RequiredVersionMaj=5700, RequiredVersionMin=0000

3. get_vse_versions
   a. Contains both required and recommended version numbers

Scheduled vs. Real Time Scores

The collection scripts run on a configured schedule and as a result not all of the data sources update events in Splunk in real time. Thus, scores are calculated on a schedule as opposed to real time. Since this app is looking to maintain awareness of general health and status as opposed to alerting on new events this schedule is reasonable but should be configured for the user's convenience and while the default is once a day, a longer interval would still be reasonable.

Measurement Methods

Mitigations would ideally be measured by observing their effects. SAMI only measures effects for the PtH metric and null page mapping in AE; the rest of the metrics focus on the method of implementation for the mitigations. Thus, if an organization has implemented a different, but functionally equivalent mitigation SAMI will not recognize the alternative and the organization will score poorly. This version of SAMI depends on using McAfee Virus Scan Enterprise (VSE) for AVFRS, McAfee HIPS for HIPS, Microsoft AppLocker or Software Restriction Policies for AW, and Microsoft EMET for AE. If an organization does not intend to implement the software in a given metric then that metric should be deactivated by turning off related scheduled searches.

Appendix A

Installation and Configuration Guide

The following is a guide to assist with the installation of SAMI into Splunk, as well as with the configuration of its customizable components.

Installation Prerequisites

Software Requirements

1. Windows Operating System
2. PowerShell 3.X or higher, so that scripts implicitly load the AD module for PowerShell.
3. Active Directory module for PowerShell. This module is included in the RSAT (Remote Server Administration Tools) feature. The only cmdlets used by SAMI are get-adgroup, get-adgroupmember, and get-adcomputer. More information can be found at https://technet.microsoft.com/en-us/library/dd378937(WS.10).aspx.
4. Splunk 6 or higher

Installation

The Splunk app should be installed prior to any ingestion of data (i.e., installing the deployment app).

SAMI Splunk App

Create one index: sami_summary. Once the index is created, configure the permissions so that the desired users or groups read capabilities. If your indexer and Search Head are separate then you will also need to create the index on the search head. The SAMI Splunk App should not require any additional configuration and can simply be placed within the directory **$SPLUNK_HOME\etc\apps** or installed using the SplunkWeb GUI. The hosting Splunk server(s) needs to be restarted to activate the app.

SAMI Splunk Deployment App (SAMI-TA)

The SAMI Splunk Deployment App should be moved to the deployment-apps directory on the Deployment Server (**$SPLUNK_HOME\etc\deployment-apps**). The app should then be added via the GUI in the desired server class. If a deployment server is not used, the app should be unzipped and placed in the apps directory on the desired Universal Forwarders (**$SPLUNK_HOME\etc\apps**).

Configuration

Both the SAMI deployment and SAMI Splunk apps should work automatically without configuration. However, there are a few user preferences that can be modified relatively easily.

Altering the Lookup Tables

The **$SPLUNK_HOME\etc\apps\sami\lookups** folder contains the various lookup tables mentioned in the Lookups section. To alter any of these lists, open the file in a text editor and save the changes; the lookup tables will be automatically updated.

Altering search scheduling

Each measure search is scheduled to run, and populate the summary index, once per day. To change how often the searches run, modify the scheduled runtime either from the Reports page or via

“Settings/Searches, reports, and alerts”. Note that some of the searches can take as much as 20 minutes to finish, so setting them to run more often than that is likely to cause erroneous results. Additionally, the “{metric}_MISSING_DATA” searches must be scheduled such that they begin only after all other searches for that metric have finished.

User Permissions

Please remember that the Splunk ‘Role’ for the user running the app should be set to search against relevant indexes by default (i.e., sami_summary). Splunk administrators can set this configuration for the appropriate Splunk Role(s) in the "Indexes Searched by Default" configuration. This configuration is located under ‘Settings » Access controls » Roles »’<RoleName>’ in Splunk Web.

Appendix B

TA-SAMI

inputs.conf – lists all required data collection for endpoints.

wmi.conf – lists required wmi queries for endpoints.

Scripts

ae.exe – Compiled Python script used to collect data required for AE checks that cannot be collected by native Splunk capabilities. Regular Python script included in the app for reference.

av.exe – Compiled Python script used to collect data required for AVFRS checks that cannot be collected by native Splunk capabilities. Regular Python script included in the app for reference.

SAMI App

props.conf – Two extractions to gather rule attributes from XML found in registry for AppLocker. One extraction for path rules and one extraction for publisher rules.

hostList – hostList.bat, run daily, output monitored. This passes a Powershell function that requires the ActiveDirectory PS module. This gets a list of all enabled hosts in Active Directory. The data is then used by SAMI to determine which hosts are active but missing data.

dcList – dcList.bat, run daily, output monitored. This passes a Powershell function that requires the ActiveDirectory PS module. This gets a list of all accounts with domain admin privileges in Active Directory. This data is used by SAMI in the PtH metric.

Searches – There is one search per measure. Each of these searches is scheduled to run daily (default). The output of these searches is stored in the sami_summary summary index. Summary dashboards use data from the last 24 hours (default) in the summary index.

Sourcetypes

WinRegistry – Regmon events.

SrpV2Rule – AppLocker rules (XML extracted from regmon).

WMI:QFE – Results from WMI QuickFixEngineering query.

WMI:OperatingSystem – Results from WMI query for operating system details.

WinHostMon – WinHostMon events.

hostList – list of host names resulting from hostList script run on the Splunk server.

dcList – list of domain admin accounts resulting from dcList script run on the Splunk server.

scriptAE – results of the AE Python script

scriptAV – results of the AV Python script

Macros

penalty_lookup – assumes the measure is set to a value and does a lookup to return the penalty for the measure.

Scoring

The first version of scoring in SAMI accumulates penalties using the following formula. This penalty is applied to measures within a metric for metric scores and across all measures for the total score (thus, the total score will not be the average of the metric scores).

$$Total\ Score = 1 + 9\ (\prod_{penalties} \frac{100 - penalty100}{})$$